

***Дэманстрацыйна-тэставая ферма праграм для платформы  
Android з вэб-інтэрфейсам  
Каваленка У.Ю., Касцюк Д.А.***

*Брэсцкі дзяржаўны тэхнічны універсітэт, [volodik666@gmail.com](mailto:volodik666@gmail.com)*

A virtualized testing farm for Android applications is presented, with substitution of default emulation tools of Google with different set of components, using nested virtualization and transparent emulation on native architecture. Presented solution provides user control via web interface, as far as typical development access through ADB. Internal architecture, deployment and scaling principles are discussed.

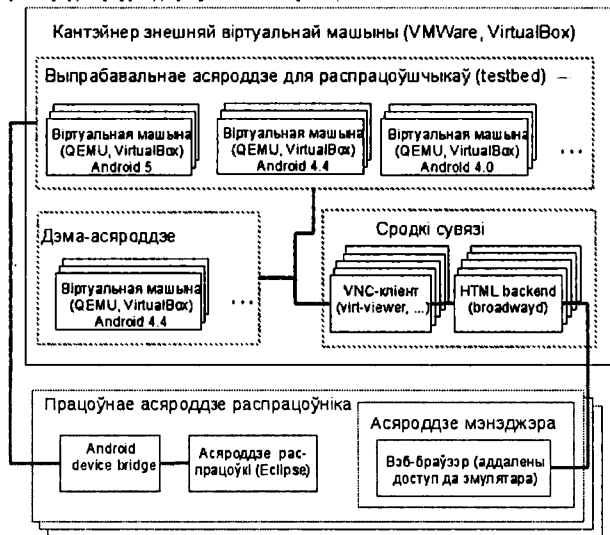
Праблему тэставання і адладкі праграмага кода ва ўмовах моцнай фрагментацыі мэтавых апаратных платформаў нельга назваць новай; аднак, на сённяшні дзень платформа Android з'яўляецца адной з найбольш фрагментаваных. Распрацоўніку неабходна трымаць на сваёй машыне набор вобразаў для эмулятара з рознымі версіямі аперацыйнай сістэмы, і запускаяць іх па чарзе. Пры гэтым, нават калі б штатныя сродкі распрацоўкі Android падавалі зручны інтэрфейс для адначасовага запуску некалькіх асобнікаў эмулятара для тэставання праграмы, нізкая прадукцыйнасць эмуляцыі, звязаная з розніцай працэсарных архітэктур, пры множным запуску эмулятара негатыўна адбілася б на прадукцыйнасці і эфектыўнасці распрацоўкі.

Пры калектыўнай распрацоўцы часам аказваецца, што метады лакальнай эмуляцыі зусім не падыходзяць. Перыядычна патрабуецца калек-

тыўны доступ да экзэмпляра тэставага прыкладання, а таксама і дэманстрацыйны доступ да прамежкавых зборак.

Відавочным выхадам з'яўляецца разгортванне тэставай фермы на сэрвэры і доступ да яе па сетцы. Дзякуючы выкарыстанню адкрытых (і узаемазамальных) сродкаў распрацоўкі, пабудова альтэрнатыўнай інфраструктуры дае такія перавагі, як эканомія вылічальных рэсурсаў машын распрацоўшчыкаў, магчымасць цэнтралізаванага аўтаматычнага кіравання эмулятарамі і дэманстрацыі працы запушчаных у іх праграм, а таксама, для шэрагу задач, замена штатнага эмулятара больш прадукцыйнай віртуальнай машынай, якая ажыццяўляе натывуымі эмуляцыю апаратнай платформы, ідэнтычнай з хост-сістэмай. Ніжэй мы ўяўляем варыянт арганізацыі такога рашэння (малюнак 1), распрацаваны для патрэб канкрэтнага прадпрыемства, аднак досыць універсальны для таго, каб прадстаўляць больш шырокую практычную цікавасць.

Для зручнасці разгортвання фермы і яе міграцыі набор эмулятараў працуе ўнутры кантэйнера віртуальнай машыны на сэрвэры лакальнай сеткі. Сістэмай знешняй віртуалізацыі можа быць альбо VirtualBox, альбо камерцыйная VMware, у залежнасці ад ужо існуючай інфраструктуры прадпрыемства, у якую павінна быць інтэграваная ферма (сумяшчэнне дзвюх сістэм віртуалізацыі ў сеткі можа апынуцца неметаэагодным па арганізацыйных прычынах). Лік ўкладзеных віртуальных машын (эмулятараў) вар'іруецца ў працэсе эксплуатацыі.

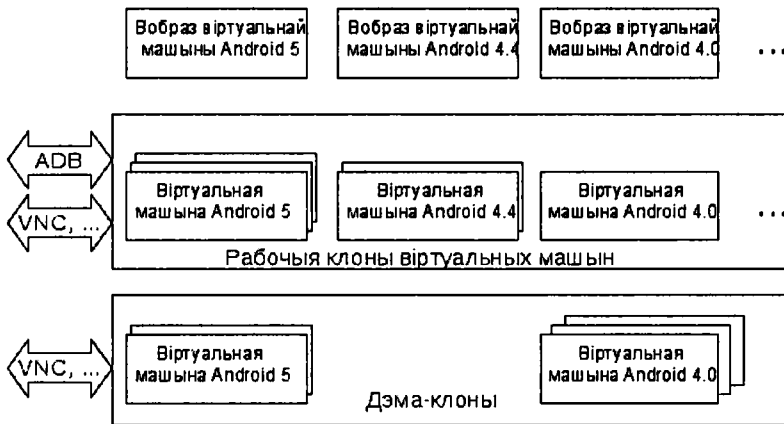


Малюнак 1. Архітэктара фермы

У якасці ўкладзенай віртуальнай машыны ў прапанаваным вырашэнні можна выкарыстоўваць QEMU або VirtualBox (выбар залежыць ад знешняй віртуальнай машыны, з якой павінна працаваць апаратна-

паскораная укладзеная віртуалізацыя). Як ужо згадвалася, штатны эмулятар з Android SDK мае праблемы з прадукцыйнасцю, менш падыходзіць для асобнага разгортвання і выкарыстання ў дэманстрацыйных мэтах. Па гэтай прычыне некаторыя распрацоўшчыкі выкарыстоўваюць зборкі Android для x86-сістэм [3], запусчаныя ў адной з натывуных сістэм віртуалізацыі без эмулявання іншай працэсарнай архітэктуры. Такі падыход захоўвае ўсе перавагі і выгоды, якія прадстаўляюцца сервернымі і дэсктопнымі сістэмамі віртуалізацыі па частцы аўтаматызацыі, кланавання і ўбудавання ў інфраструктуру прадпрыемства.

У залежнасці ад таго, што выкарыстоўваецца ў якасці укладзенай віртуальнай машыны, доступ карыстальнікаў (як непасрэдна распрацоўшчыкаў, так і менеджменту) да эмулятара можа ажыццяўляцца праз пратакол VNC, SPICE альбо RDP (акрамя таго, што для распрацоўшчыкаў прадугледжаны доступ да эмулятара праз ADB). Першапачаткова для злучэння з эмулятарам праз вэб-сераду мы планавалі выкарыстанне аддаленых кліентаў на JavaScript, якія паспяхова прымяняюцца намі ў іншых праектах [1, 2] - такіх як noVCN, Guacamole, FreeRDP-webview. Аднак у працэсе тэставання быў зроблены выбар вэб-доступу праз GTK-бэкэнд broadway. На гэта паўплывала лепшая прадукцыйнасць C-кода, адсутнасць неабходнасці ў дадатковых кампанентах, якія трансліюць злучэнне TCP ў вэб-сокеты, лёгкасць ўзняцця новага злучэння і доступу да яго.



Малюнак 2. Структура кіравання віртуальнымі машынамі

Структура кіравання кампанентамі сістэмы выглядае наступным чынам (малюнак 2). На баку кліента доступ да эмулятара (як на машынах распрацоўшчыкаў, так і кіраўнікоў) ажыццяўляецца праз вэб-браўзэр, які ўзаемадзейнічае з GTK-бакэндам broadway, а broadway малюе вокны выбраных GTK3-прыкладанняў на канвасе HTML5. Праграма на GTK

запускаецца ў headless-рэжыме, то ёсць толькі для сеткавага доступу. Для гэтага ў сістэме запускаюцца дэмань broadwayd, кожны з якіх займае асобны сеткавай порт, а праграме пры запуску перадаецца дадатковы набор параметраў каманднага радка, якія вызначаюць рэжым адмалёўкі для бібліятэкі GTK.

Распрацоўшчык ўзаемадзейнічае з укладзенай віртуальнай машынай праз стандартны інтэрфейс ADB, а тажа просты вэб-інтэрфейс, які дазваляе кланаваць адну з ўкладзеных віртуальных машын.

Маюцца эталонныя вобразы віртуальных машын з Android-x86 розных версій. Яны сканфігураваны і настроены для максімальна хуткага запуску новай машыны. Для тэставання распрацоўванага ПА клануецца адзін з эталонных вобразаў. Разгортванне і адладка праграмы ў кланаванай машыну выконваецца праз ADB, а кіраванне - праз вэб інтэрфейс. Для стварэння ўнутраных дэма-версій патрэбныя вобразы дадаткова клануюцца для дэма-асяроддзя, што працуе таксама праз вэб інтэрфейс.

### **Літаратура**

1. Касцюк Д.А., Луцюк П.А., Уласенка С.С., Жалудок В.А. Ужыванне віртуальных машын у складзе ілюстраваных аглядаў гісторыі праграмнага забеспячэння // Трэця міжнародна навуковапрактычна конференцыя FOSS Lviv 2014: Збірнік навуковых праць / Львів, 24–27 квітня 2014 р. – С. 51–54.
2. Коваленко В.Ю., Костюк Д.А., Кричко А.Г. Подход к автоматизации распределенного тестирования графических приложений // Информационные технологии и системы 2014 (ИТС 2014): материалы международной научной конференции. Минск, БГУИР, 29.10.2014. – С. 104–105.
3. Android-x86 Project – Run Android on Your PC. <http://www.android-x86.org/>